

LAPORAN PRAKTIKUM
MODUL 5
“SINGLE LINKED LIST (BAGIAN KEDUA)”



Disusun oleh:

Wendri Tri Pambudi (21104048 / SE07-01)

Dosen:

Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM PURWOKERTO

2024

Tugas Pendahuluan

1. Soal 1: Mencari Elemen Tertentu dalam SLL

Deskripsi Soal: Buatlah program yang memungkinkan pengguna memasukkan 6 elemen integer ke dalam list. Implementasikan function `searchElement` untuk mencari apakah sebuah nilai tertentu ada dalam list.

Instruksi

1. Minta pengguna untuk memasukkan nilai yang ingin dicari.
2. Jika nilai ditemukan, tampilkan alamat dan posisi dalam angka (contoh: urutan ke 4) pada list tersebut.
3. Jika nilai tidak ditemukan, tampilkan pesan bahwa elemen tersebut tidak ada dalam list tersebut.

NB:

1. Gunakan pendekatan linier search untuk mencari elemen.

Code Soal 1:

```
05_Single_Linked_List_Bagian_2 > TP > C++ SOAL_01.cpp > main()
3
4 struct Node {
5     int data;
6     Node* next;
7 };
8
9 void insertNode(Node* &head, int value) {
10     Node* newNode = new Node();
11     newNode->data = value;
12     newNode->next = NULL;
13
14     if (head == NULL) {
15         head = newNode;
16     } else {
17         Node* temp = head;
18         while (temp->next != NULL) {
19             temp = temp->next;
20         }
21         temp->next = newNode;
22     }
23 }
24
25 // Fungsi untuk mencari elemen dalam list
26 void searchElement(Node* head, int value) {
27     Node* current = head;
28     int position = 1;
29
30     while (current != NULL) {
31         if (current->data == value) {
32             cout << "Elemen " << value << " ditemukan di alamat " << current << " pada urutan ke-" << position << "." << endl;
33             return;
34         }
35         current = current->next;
36         position++;
37     }
38
39     cout << "Elemen " << value << " tidak ditemukan dalam list." << endl;
40 }
41
42 int main() {
43     Node* head = NULL;
44
45     cout << "Masukkan 6 elemen ke dalam list:" << endl;
46     for (int i = 0; i < 6; i++) {
47         int value;
48         cout << "Elemen " << i+1 << ": ";
49         cin >> value;
50         insertNode(head, value);
51     }
52
53     int cari;
54     cout << "Masukkan elemen yang ingin dicari: ";
55     cin >> cari;
56
57     searchElement(head, cari);
58
59     return 0;
60 }
61
```

Ouput Soal 1:

```
Masukkan 6 elemen ke dalam list:
Elemen 1: 4
Elemen 2: 2
Elemen 3: 5
Elemen 4: 1
Elemen 5: 3
Elemen 6: 2
Masukkan elemen yang ingin dicari: 5
Elemen 5 ditemukan di alamat 0x7e9ee0 pada posisi ke-3.
```

2. Mengurutkan List Menggunakan Bubble Sort

Deskripsi Soal: Buatlah program yang mengizinkan pengguna memasukkan 5 elemen integer ke dalam list. Implementasikan procedure bubbleSortList untuk mengurutkan elemen-elemen dalam list dari nilai terkecil ke terbesar.

Instruksi

Setelah mengurutkan, tampilkan elemen-elemen list dalam urutan yang benar.

Langkah-langkah Bubble Sort pada SLL

1. Inisialisasi:

- -Buat pointer current yang akan digunakan untuk menelusuri list.
- Gunakan variabel boolean swapped untuk mengawasi apakah ada pertukaran yang dilakukan pada iterasi saat ini.

2. Traversing dan Pertukaran:

- Lakukan iterasi berulang sampai tidak ada pertukaran yang dilakukan:
 - Atur swapped ke false di awal setiap iterasi.
 - Set current ke head dari list.
 - Selama current.next tidak null (masih ada node berikutnya):
 - Bandingkan data pada node current dengan data pada node current.next.
 - Jika data pada current lebih besar dari data pada current.next, lakukan pertukaran:
 - Tukar data antara kedua node (bukan pointer).
 - Set swapped menjadi true untuk menunjukkan bahwa ada pertukaran yang dilakukan.
 - Pindahkan current ke node berikutnya (current = current.next).

3. Pengulangan:

- Ulangi langkah 2 sampai tidak ada lagi pertukaran yang dilakukan (artinya list sudah terurut).

Code Soal 2:

```

05_Single_Linked_List_Bagian_2 > TP > C++ SOAL_02.cpp > printList
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  void insertNode(Node* &head, int value) {
10     Node* newNode = new Node();
11     newNode->data = value;
12     newNode->next = NULL;
13
14     if (head == NULL) {
15         head = newNode;
16     } else {
17         Node* temp = head;
18         while (temp->next != NULL) {
19             temp = temp->next;
20         }
21         temp->next = newNode;
22     }
23 }
24
25 void printList(Node* head) {
26     Node* temp = head;
27     while (temp != NULL) {
28         cout << temp->data << " ";
29         temp = temp->next;
30     }
31     cout << endl;
32 }

```

```

void bubbleSortList(Node* head) {
    if (head == NULL) return;

    bool swapped;
    Node* current;
    Node* lastSorted = NULL;

    do {
        swapped = false;
        current = head;

        while (current->next != lastSorted) {
            if (current->data > current->next->data) {
                int temp = current->data;
                current->data = current->next->data;
                current->next->data = temp;
                swapped = true;
            }
            current = current->next;
        }
        lastSorted = current;
    } while (swapped);
}

int main() {
    Node* head = NULL;

    cout << "Masukkan 5 elemen ke dalam list:" << endl;
    for (int i = 0; i < 5; i++) {
        int value;
        cout << "Elemen " << i+1 << ": ";
        cin >> value;
        insertNode(head, value);
    }

    cout << "List sebelum diurutkan: ";
    printList(head);

    bubbleSortList(head);

    cout << "List setelah diurutkan: ";
    printList(head);

    return 0;
}

```

Output Soal 2:

```

Masukkan 5 elemen ke dalam list:
Elemen 1: 2
Elemen 1: 2
Elemen 2: 1
Elemen 3: 0
Elemen 4: 4
Elemen 5: 8
List sebelum diurutkan: 2 1 0 4 8
List setelah diurutkan: 0 1 2 4 8

```

3. Menambahkan Elemen Secara Terurut

Deskripsi Soal: Buatlah program yang mengizinkan pengguna memasukkan 4 elemen integer ke dalam list secara manual. Kemudian, minta pengguna memasukkan elemen tambahan yang harus ditempatkan di posisi yang sesuai sehingga list tetap terurut.

Instruksi

1. Implementasikan procedure insertSorted untuk menambahkan elemen baru ke dalam list yang sudah terurut.
2. Tampilkan list setelah elemen baru dimasukkan.

Code Soal 3:

```
05_Single_Linked_List_Bagian_2 > TP > C++ SOAL_03.cpp > printList(Node *)
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  void insertSorted(Node* &head, int value) {
10     Node* newNode = new Node();
11     newNode->data = value;
12     newNode->next = NULL;
13
14     if (head == NULL || head->data >= value) {
15         newNode->next = head;
16         head = newNode;
17     } else {
18         Node* current = head;
19         while (current->next != NULL && current->next->data < value) {
20             current = current->next;
21         }
22         newNode->next = current->next;
23         current->next = newNode;
24     }
25 }
26
27 void printList(Node* head) {
28     Node* temp = head;
29     while (temp != NULL) {
30         cout << temp->data << " ";
31         temp = temp->next;
32     }
33     cout << endl;
34 }
35
```

```
05_Single_Linked_List_Bagian_2 > TP > C++ SOAL_03.cpp > main()
35
36 int main() {
37     Node* head = NULL;
38
39     cout << "Masukkan 4 elemen (secara terurut):" << endl;
40     for (int i = 0; i < 4; i++) {
41         int value;
42         cout << "Elemen " << i+1 << ": ";
43         cin >> value;
44         insertSorted(head, value);
45     }
46
47     cout << "List setelah memasukkan 4 elemen: ";
48     printList(head);
49
50     int additionalValue;
51     cout << "Masukkan elemen tambahan yang ingin ditambahkan: ";
52     cin >> additionalValue;
53
54     insertSorted(head, additionalValue);
55
56     cout << "List setelah menambahkan elemen baru: ";
57     printList(head);
58
59     return 0;
60 }
61
```

Output Soal 3:

```
Masukkan 4 elemen (secara terurut):
Elemen 1: 5
Elemen 2: 8
Elemen 3: 2
Elemen 4: 5
List setelah memasukkan 4 elemen: 2 5 5 8
Masukkan elemen tambahan yang ingin ditambahkan: 3
List setelah menambahkan elemen baru: 2 3 5 5 8
```